## Claims

What is claimed is:

1. A method of storing a flattened structured data document, comprising the steps of:

a) receiving the flattened structured data document having a plurality of lines, each of the lines having a tag, a data entry and a format character;

b) storing the tag in a dictionary store;

c) storing the data entry in a dictionary store; and

d) storing the format character, a tag dictionary offset and a data dictionary offset in a map store.

2. The method of claim 1, wherein step (b) further includes the steps of:

b1) transforming the tag to form a tag transform;

b2) storing the tag dictionary offset in a dictionary index at an address pointed to by the tag transform.

3. The method of claim 1, wherein step (c) further includes the steps of:

c1) transforming the data entry to form a data transform;

c2) storing the data dictionary offset in a dictionary index at an address pointed to by the data transform.

4. The method of claim 2, wherein step (b1) further includes the steps of:

i) determining if the tag is unique;

ii) when the tag is unique, storing the tag in the dictionary store;

iii) when the tag is not unique, the tag is not stored in the dictionary store.

5. The method of claim 4, wherein step (i) further includes the steps of:

determining if a tag pointer is stored in the dictionary index at an address equal to the tag transform;

when the tag pointer is stored in the dictionary index, the tag is not unique.

6. The method of claim 5, further including the step of:

when the tag pointer is not stored in the associative index, the tag is unique.

7. The method of claim 1, wherein step (a) further including the step of:

a1) wherein each of the lines have a plurality of tags.

8. The method of claim 1 further including the steps of:

e) creating a map index;

f) determining if the tag is unique;

g) when the tag is unique, storing a pointer to a map location of the tag.

9. The method of claim 8, further including the steps of:

h) when the tag is not unique, determining if a duplicates flag is set;

i) when the duplicates flag is set, incrementing a duplicates count.

10. The method of claim 9, further including the steps of:

j) when the duplicates flag is not set, setting the duplicates flag;
k) setting the duplicates count to two.

11. The method of claim 10, further including the steps of:

l) calculating a transform of the tag with an instance count to form a first instance tag transform and a second instance tag transform;

m) storing a first map pointer in the map index at an address associated with the first instance tag transform.

12. The method of claim 11, further including the step of:

n) storing a second map pointer in the map index at an address associated with the second instance tag transform.

13. The method of claim 9, further including the steps of:

j) calculating a transform of the tag with an instance count equal to the duplicates count to form a next instance tag transform;

k) storing a next map pointer in the map index at an address associated with the next instance tag transform.

14. The method of claim 1 further including the steps of:

e) creating a map index;

f) determining if the data entry is unique;

g) when the data entry is unique, storing a pointer to a map location of the tag.

15. A system for storing a structured data document, comprising:

a map store having a plurality of cells each containing a dictionary pointer and a format character;

a dictionary store having a plurality of tags and a plurality of data entries; and

an associative index having a plurality of addresses each of the plurality of address having an entry flag.

16. The system of claim 15, further including a flattener that converts the structured data document into a flattened structured data document, the flattener connected to the map store.

17. The system of claim 16, further including a parser parsing the flattened structured data document for a tag and a data entry.

18. The system of claim 17, further including a transform generator connected to the parser, the transform generator converting the data entry into a tag transform.

19. The system of claim 15, further including a map index that contains a dictionary pointer.

20. The system of claim 15, wherein the format character is a delete number.

21. The system of claim 15, wherein some of the plurality of addresses are associated with a tag transform.

22. The system of claim 15, wherein some of the plurality of addresses are associated with a data transform.

23. The system of claim 15, further including a plurality of format characters.

24. The system of claim 23, wherein one of the plurality of format characters indicates a first new tag in a flattened line.

25. The system of claim 23, wherein one of the plurality of format characters indicates a number of consecutive tags closed after a data entry.

26. The system of claim 23, wherein one of the plurality of format characters indicates a parent line number of a flattened line.

27. The system of claim 23, wherein one of the plurality of format characters indicates an inserted a flattened line.

28. The system of claim 15, wherein the dictionary store includes a data dictionary store and a tag dictionary store.

29. A method of storing a flattened structured data document, comprising the steps of:

a) flattening the structured data document to form a flattened structured data document;

b) parsing each line of the flattened structured data document for a tag;

c) determining if the tag is unique;

d) when the tag is unique, storing the tag in a dictionary store.

30. The method of claim 29, further including the steps of:

e) storing a tag dictionary offset in a map store;

f) storing a plurality of format characters in the map store.

31. The method of claim 29, further including the steps of:

e) when the tag is not unique, determining a tag dictionary offset;

f) storing the tag dictionary offset in a map store.

32. The method of claim 29, wherein step (d) further includes the steps of:

d1) transforming the tag to form a tag transform;

d2) performing an associative lookup in a dictionary index using the tag transform.

33. The method of claim 32, further including the steps of:

d3) creating a map index that has a map pointer that points to a location in the map store of the tag, wherein the map pointer is stored at an address of the map index that is associated with the tag transform.